Modeling SI/PI and EMC Problems Through Data: Deterministic, Parametric, and Stochastic Perspectives – Part 2

Paolo Manfredi

Department of Electronics and Telecommunications

Politecnico di Torino

Politecnico di Torino

EMC Group

Surrogate models

Design exploration is often unfeasible due to simulation cost and size of the design space

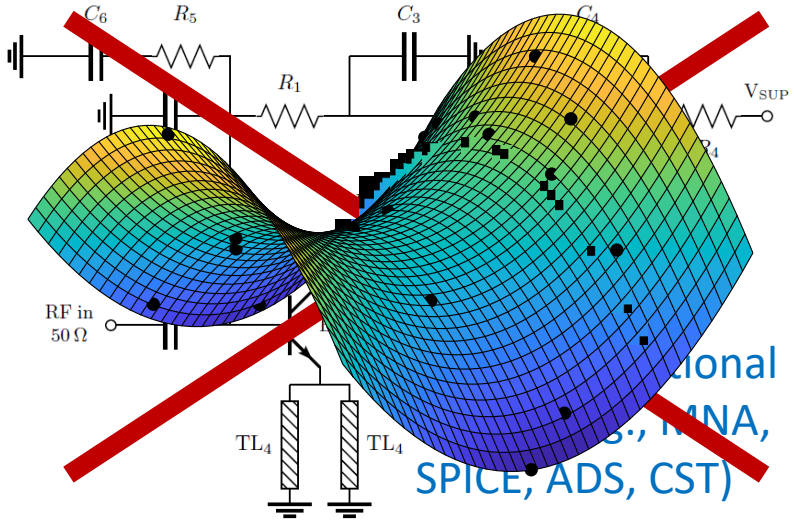
Examples: Monte Carlo analysis, yield analysis, optimization, etc.

Surrogate models are increasingly leveraged to speed-up design exploration tasks

x: input parameters

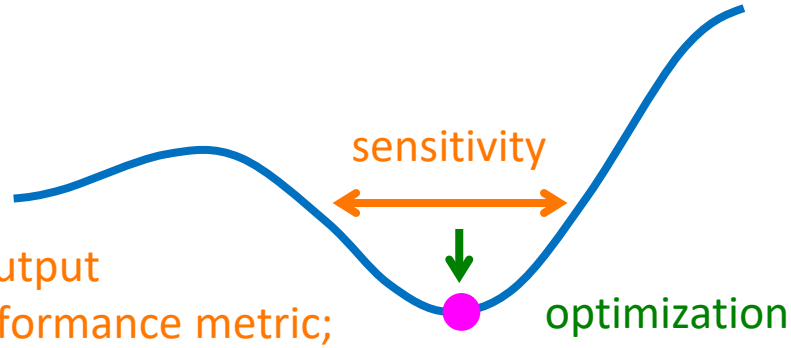
SYMBOL	MIN.	TYP.	MAX.	UNIT
V_{CBO}	–	–	10	V
V_{CEO}	–	–	4.5	V
I_C	–	25	30	mA
P_{tot}	–	–	135	mW
h_{FE}	50	80	120	
C_{re}	–	95	–	fF
f_T	–	25	–	GHz
G_{max}	–	20	–	dB
F	–	1.2	–	dB

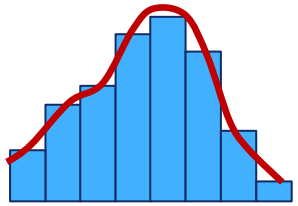
$$y = \mathcal{M}(\boldsymbol{x})$$



ional
NA,
SPICE, ADS, CST)

y: output
(performance metric;
e.g., gain)





yield

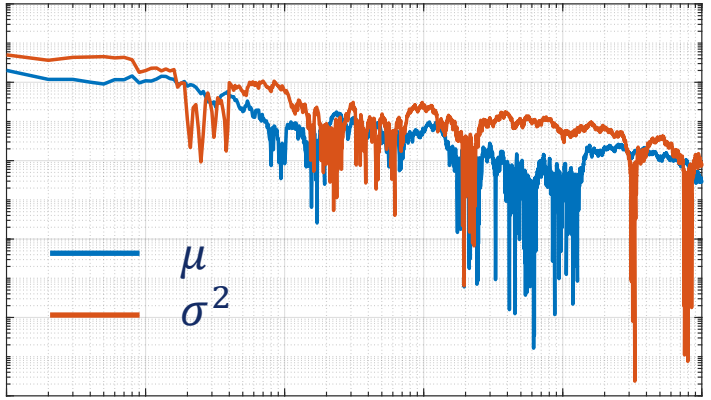
Uncertainty quantification (UQ)

Many design parameters are uncertain

Examples: fabrication tolerances, process variations, etc.

UQ: assess the impact of statistical variations on the output metrics

Statistical assessment is time consuming because slow to converge ($\sim 1/\sqrt{N}$)



Surrogate-based UQ is efficient as long as the surrogate is

1. Fast to evaluate ✅
2. Cheap to train ✅

Note: neural networks often do not meet requirement #1,
as they require massive data 💰 💰 💰 and long times ⌛ ⌛ ⌛ for training

⇒ not suitable for on-the-fly training 🪁 !

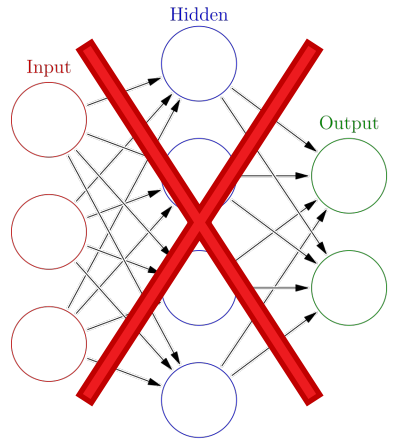


Image from Wikimedia Commons, CC BY-SA 3.0

Agenda

- Overview of Gaussian process regression (GPR)
- Application of GPR to probabilistic uncertainty quantification (UQ)
- Active learning (AL)
  - Bayesian optimization (BO)
  - UQ

Gaussian process regression (GPR)

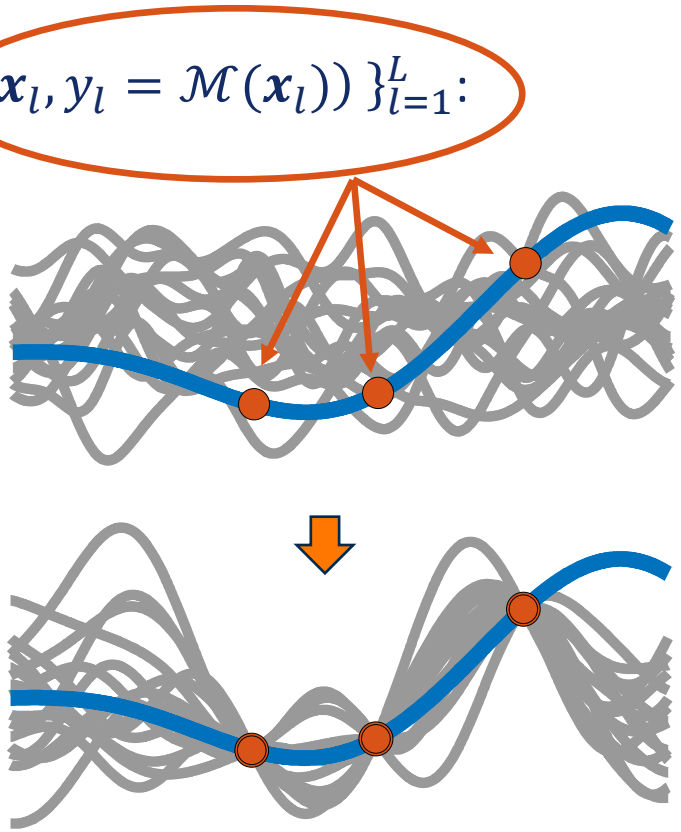
The target "map" $\mathcal{M}(\boldsymbol{x})$ is assumed to be a trajectory of a *prior* Gaussian process (GP)

The model is obtained by Bayesian inference on a set of "training samples" $\{(\boldsymbol{x}_l, y_l = \mathcal{M}(\boldsymbol{x}_l))\}_{l=1}^{L}$:

$$y \approx \hat{y} = \mathcal{M}_{\mathrm{GPR}}(\boldsymbol{x}) = \sum_{l=1}^{L} \alpha_l k(\boldsymbol{x}, \boldsymbol{x}_l)$$

Training points

where $k(\boldsymbol{x}, \boldsymbol{x}')$ is the GP covariance function or *kernel*



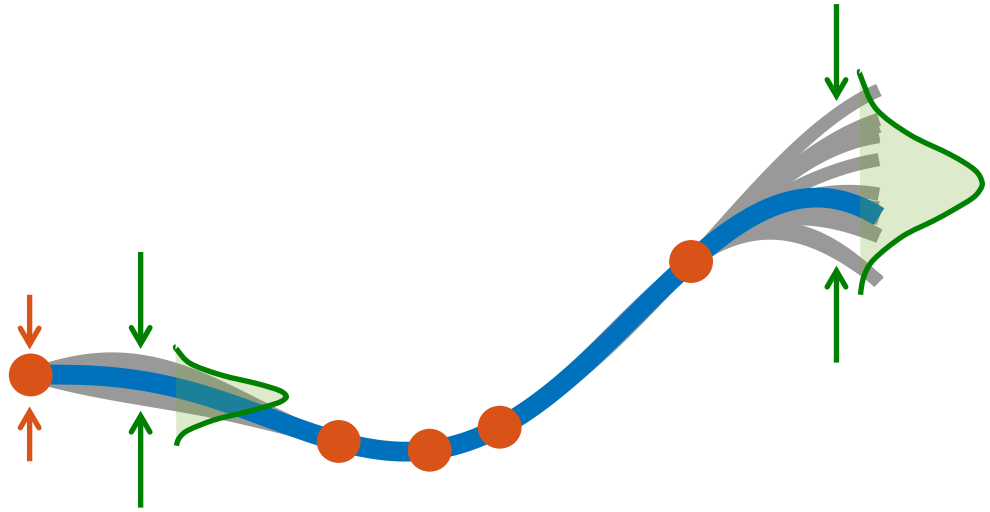
The model coefficients $\boldsymbol{\alpha}$ are computed based on the available observations

- ✅ Discard GP trajectories that are not consistent with observed data
- ✅ Model complexity is determined by the available samples L, regardless of the input dimensionality

Prediction confidence

A prediction confidence is associated to model predictions

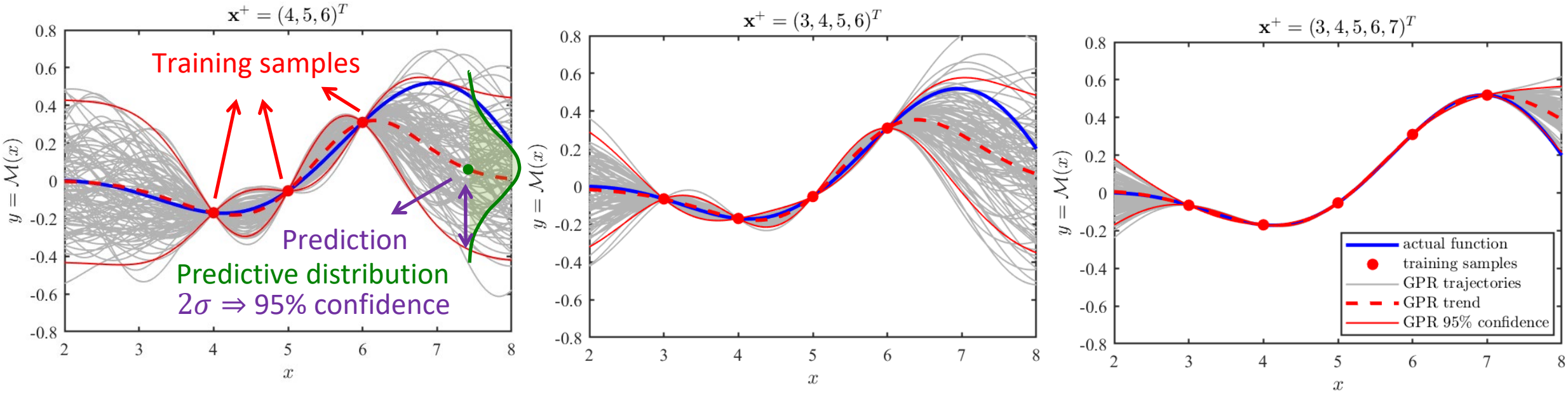
- ✅ Model prediction is a Gaussian random variable (non-deterministic!)
- ✅ Accounts for limited data availability
- ✅ Allows associating confidence levels to model predictions
- ✅ Adding new points reduces uncertainty



Illustrative 1D example

Consider the 1D analytical function (underdamped circuit): $y = \mathcal{M}(x) = e^{-10/x}(2\cos x + \sin x)$

Function approximation:



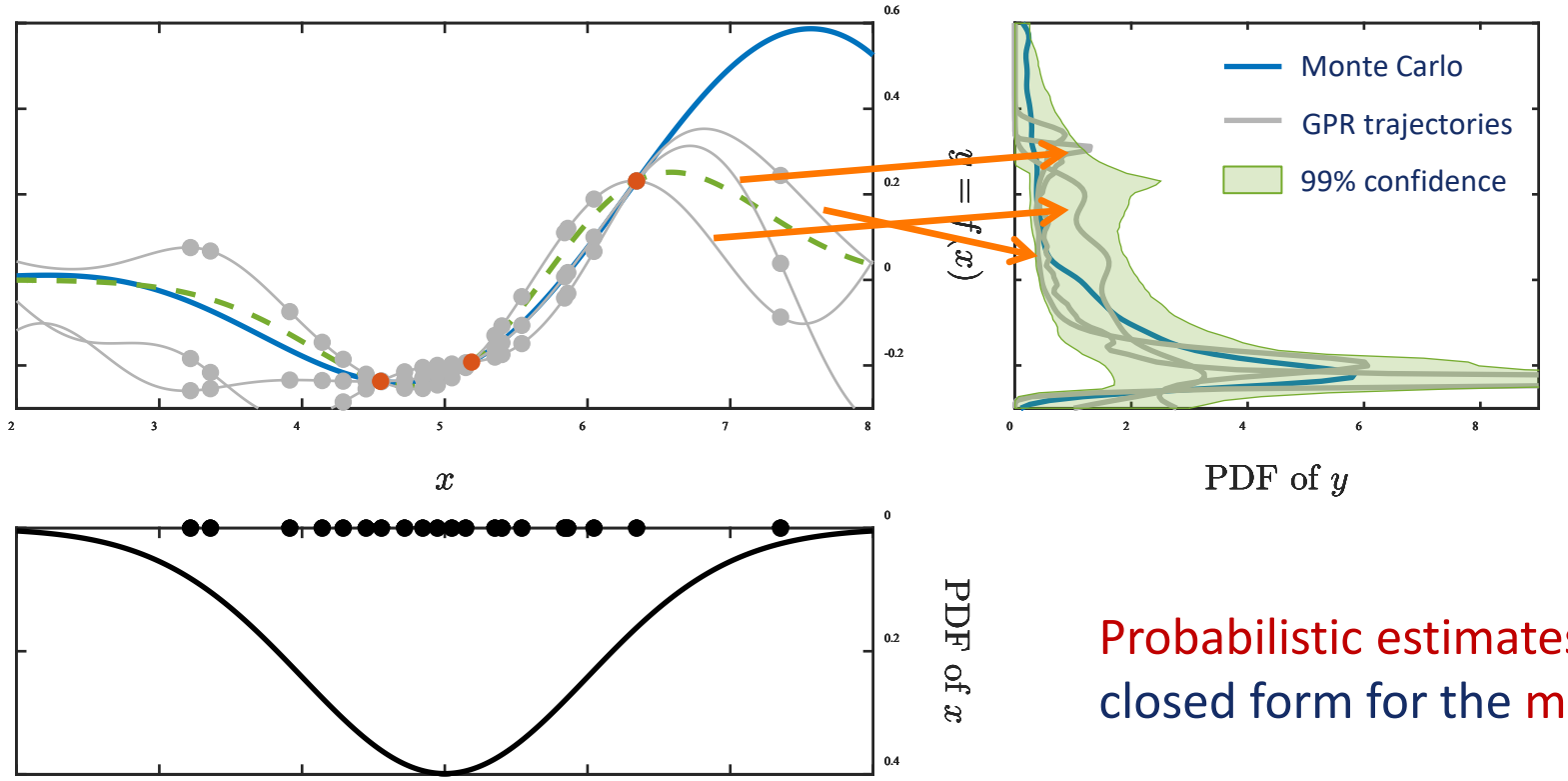
✅ Model interpolates training data

✅ Noise can be added to observations to relax exact interpolation

Monte Carlo-based UQ with GPR models

The GPR model is now used as a surrogate of $\mathcal{M}(x)$ in a Monte Carlo (MC) analysis, with $x \sim \mathcal{N}(5,1)$

Naïve approach: sample posterior trajectories with MC samples $\{x_i\}_{i=1}^{N}$!

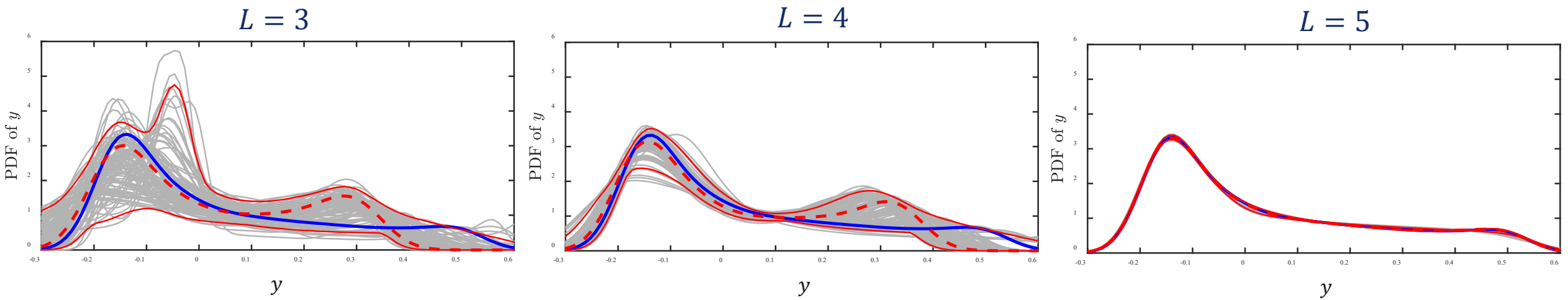


- For each trajectory, a different prediction of the MC samples is obtained
- By considering many trajectories, the **dispersion** of predictions can be estimated!

Probabilistic estimates can be also obtained in closed form for the mean and variance [1], [2]

Prediction of moments and distribution

Probability density function (PDF):

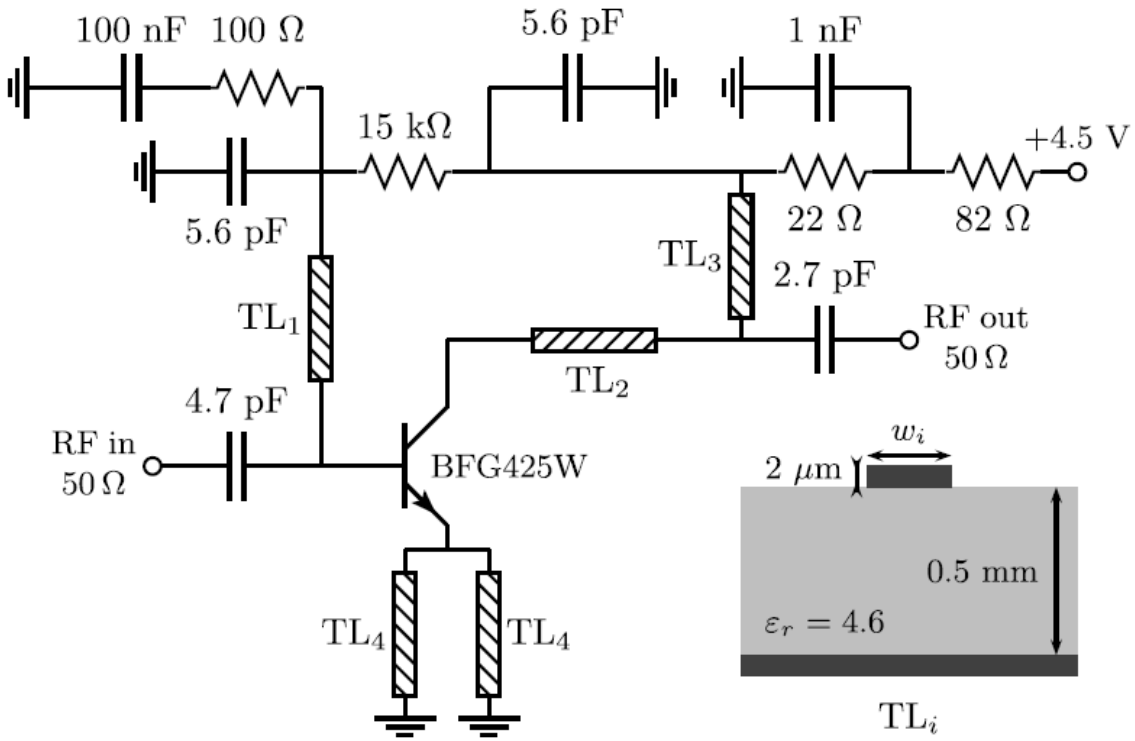


Moments (mean and variance):

Moment	Exact	GPR with 3 training samples	GPR with 4 training samples	GPR with 5 training samples
Mean	**0.0381**	[$\underline{-0.0123}$ **0.0185** $\overline{0.0492}$]	[$\underline{0.0064}$ **0.0243** $\overline{0.0422}$]	[$\underline{0.0382}$ **0.0391** $\overline{0.0401}$]
Variance	**0.0448**	[$\underline{0.0232}$ **0.0377** $\overline{0.0522}$]	[$\underline{0.0269}$ **0.0365** $\overline{0.0462}$]	[$\underline{0.0447}$ **0.0453** $\overline{0.0460}$]

A more interesting engineering example…

A 2-GHz BJT low-noise amplifier

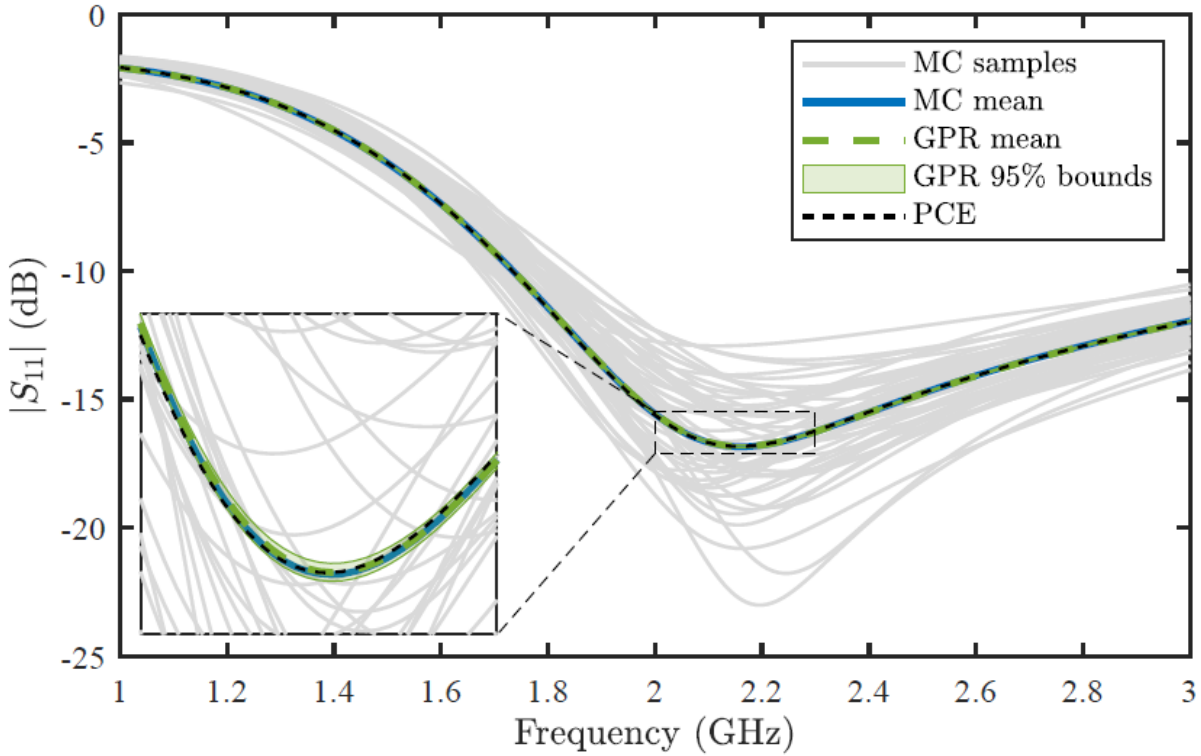


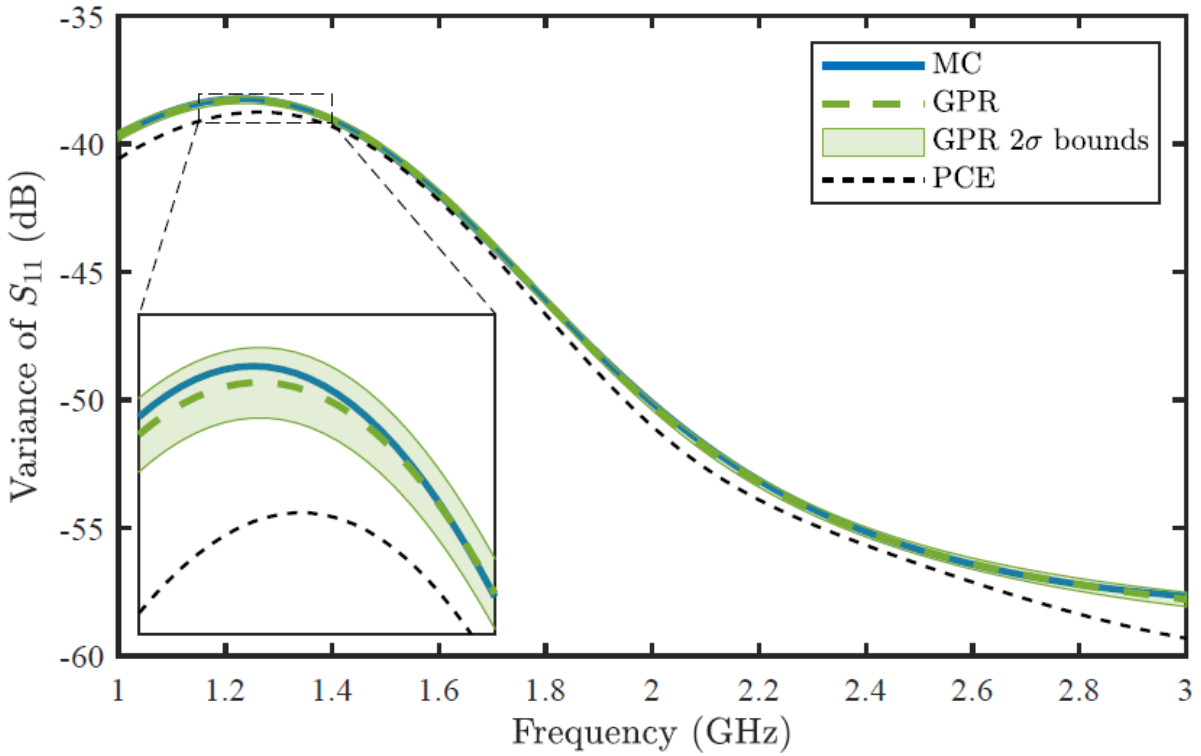
25 uncertain parameters

Lumped elements
Parasitics
BJT parameters
Mstrip parameters
….

Results (return loss)

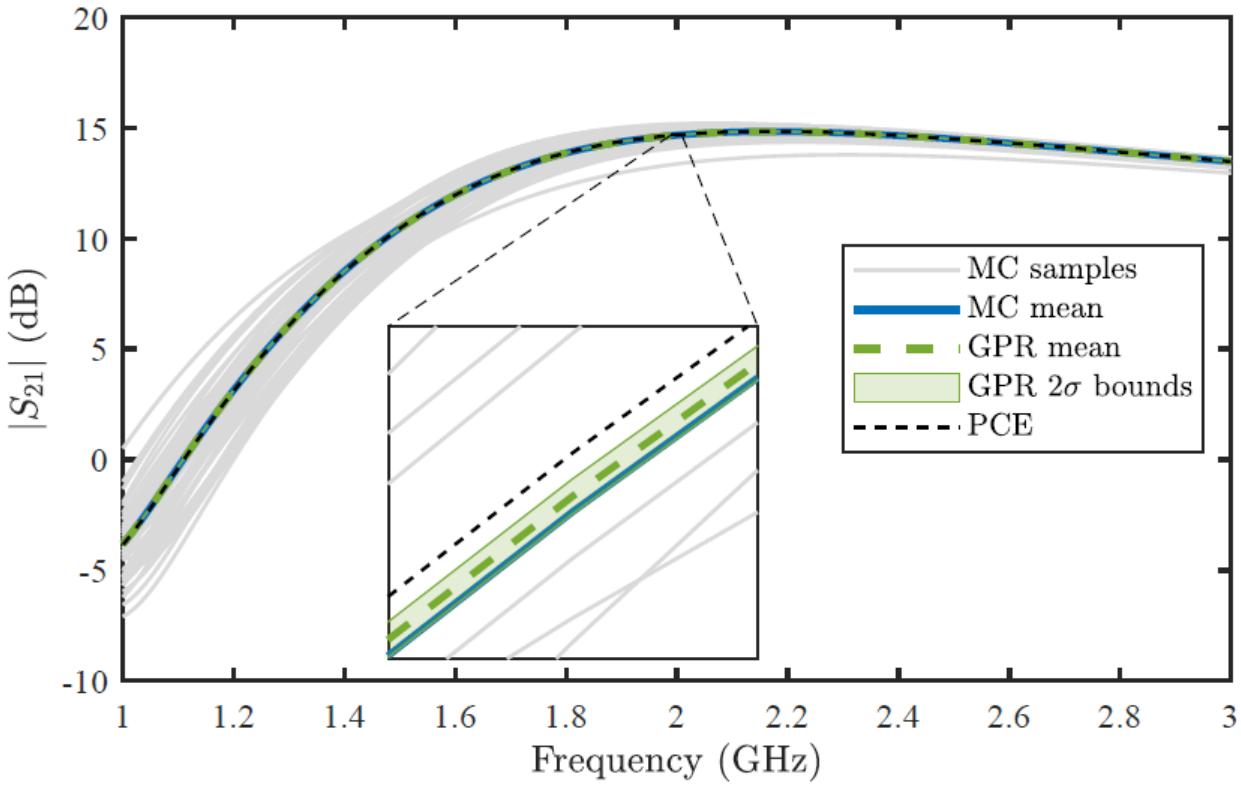
S_{11} in the frequency band [1GHz , 3GHz]

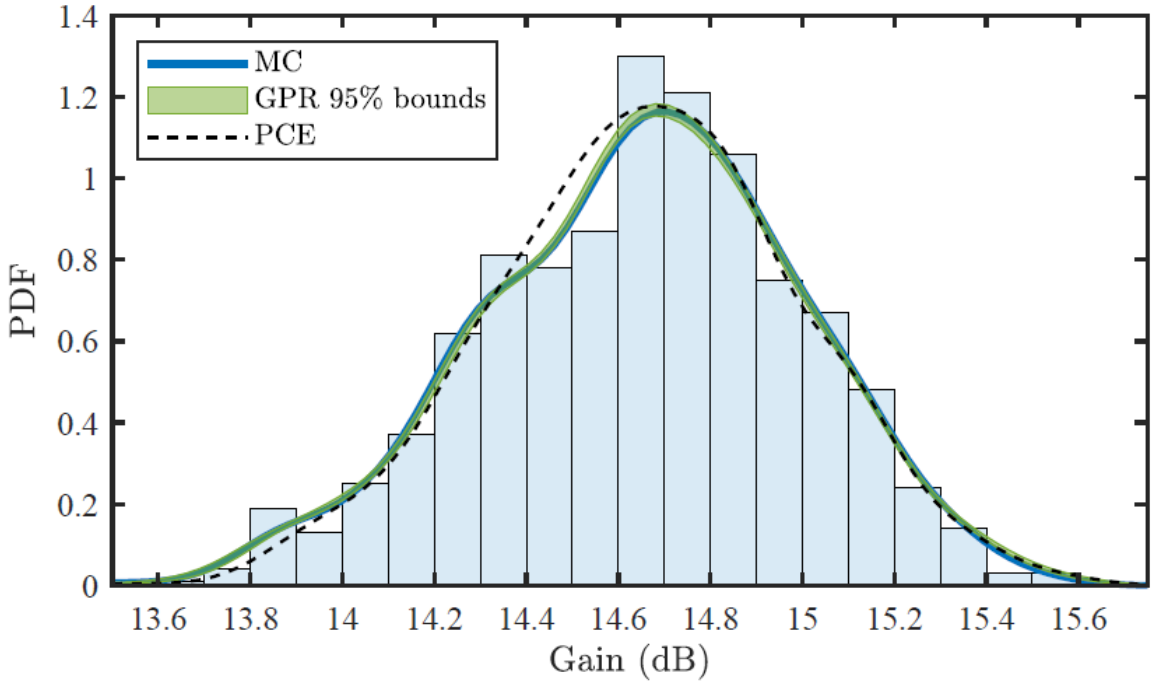




Results (gain)

Amplifier gain (S_{21})





# Training samples:	**100**
Self training time:	**20 s**

Active learning

The prediction confidence is leveraged to drive the acquisition of training points in active learning (AL) schemes

Indeed, it suggests the regions where the GPR model is less accurate (i.e., more uncertain)

- AL seeks to maximize an acquisition function (AF) to select the next training point to query
- The evaluation of the AF should be inexpensive compared to the actual computational model
- The optimal choice of the AF depends on the goal for which the GPR model is used!

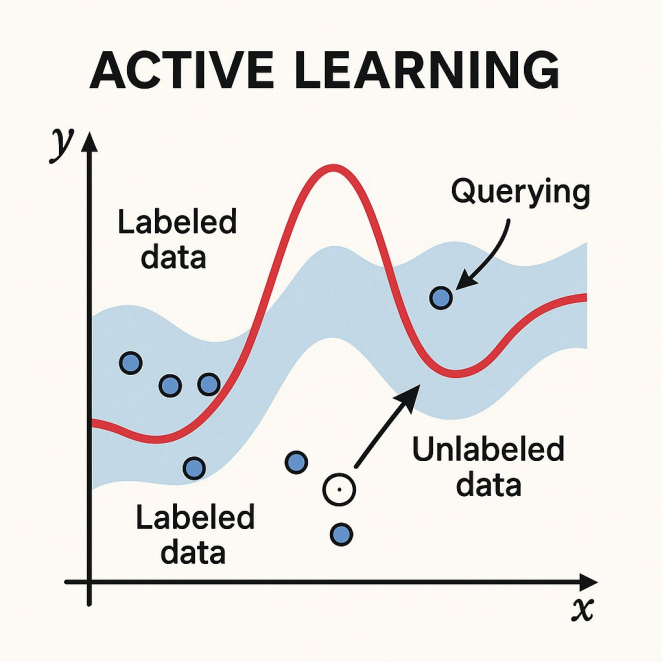
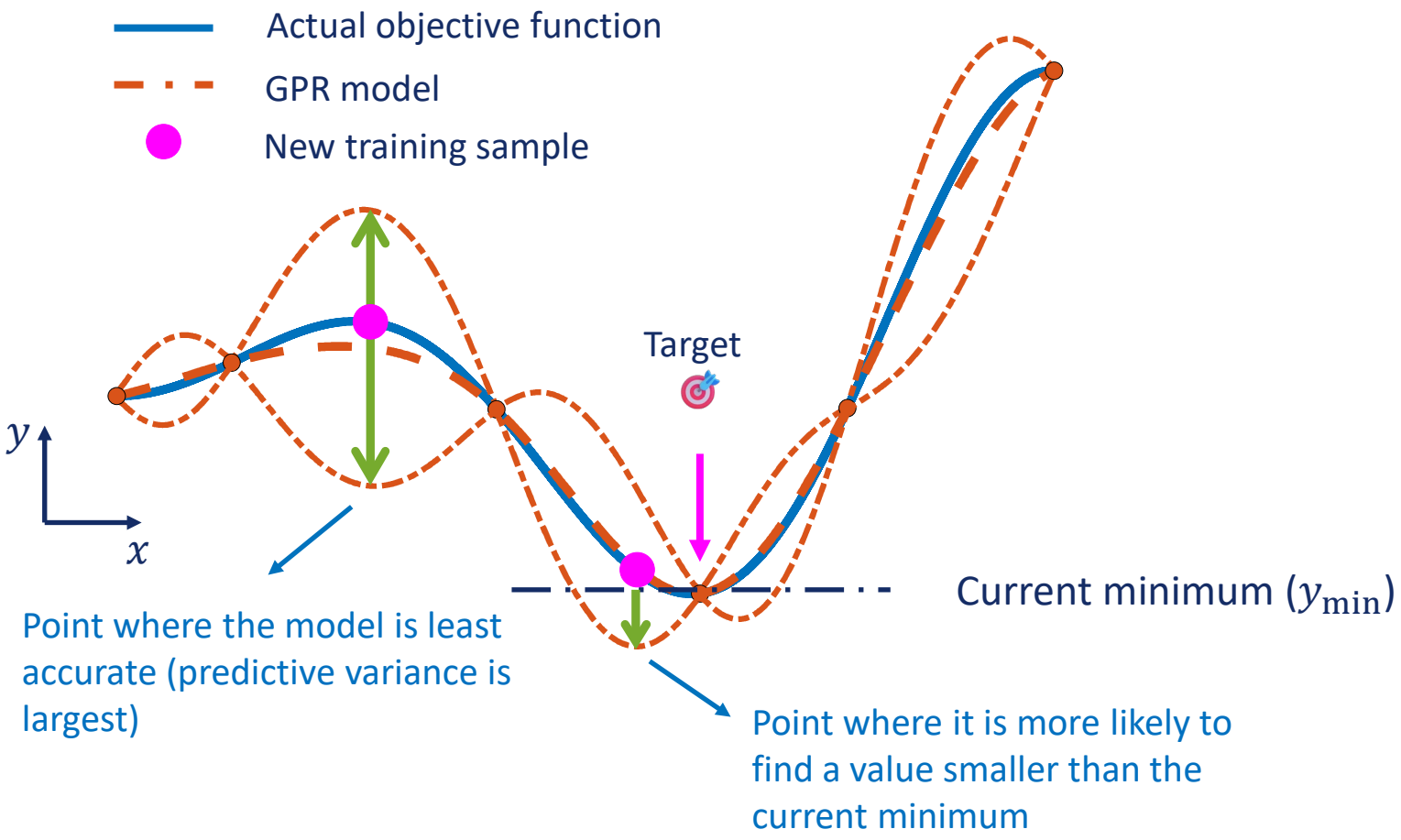


Image generate with ChatGPT 4o using DALL-E

Bayesian optimization

In Bayesian optimization (BO), a GPR model is fitted to an objective function in the process of minimizing it



Acquisition functions

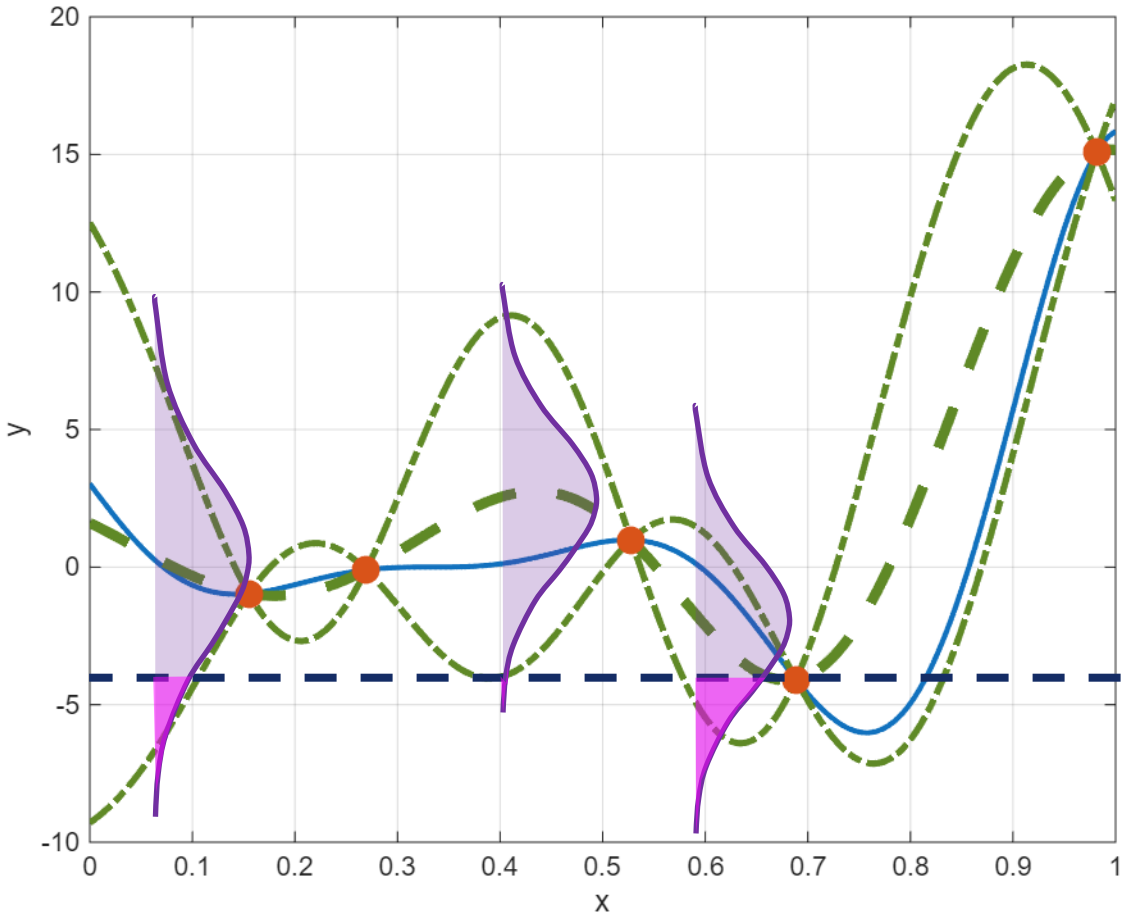
Since we know the predictive distribution (Gaussian), we can compute the probability that the prediction takes a value smaller than the current minimum!

Probability of improvement:

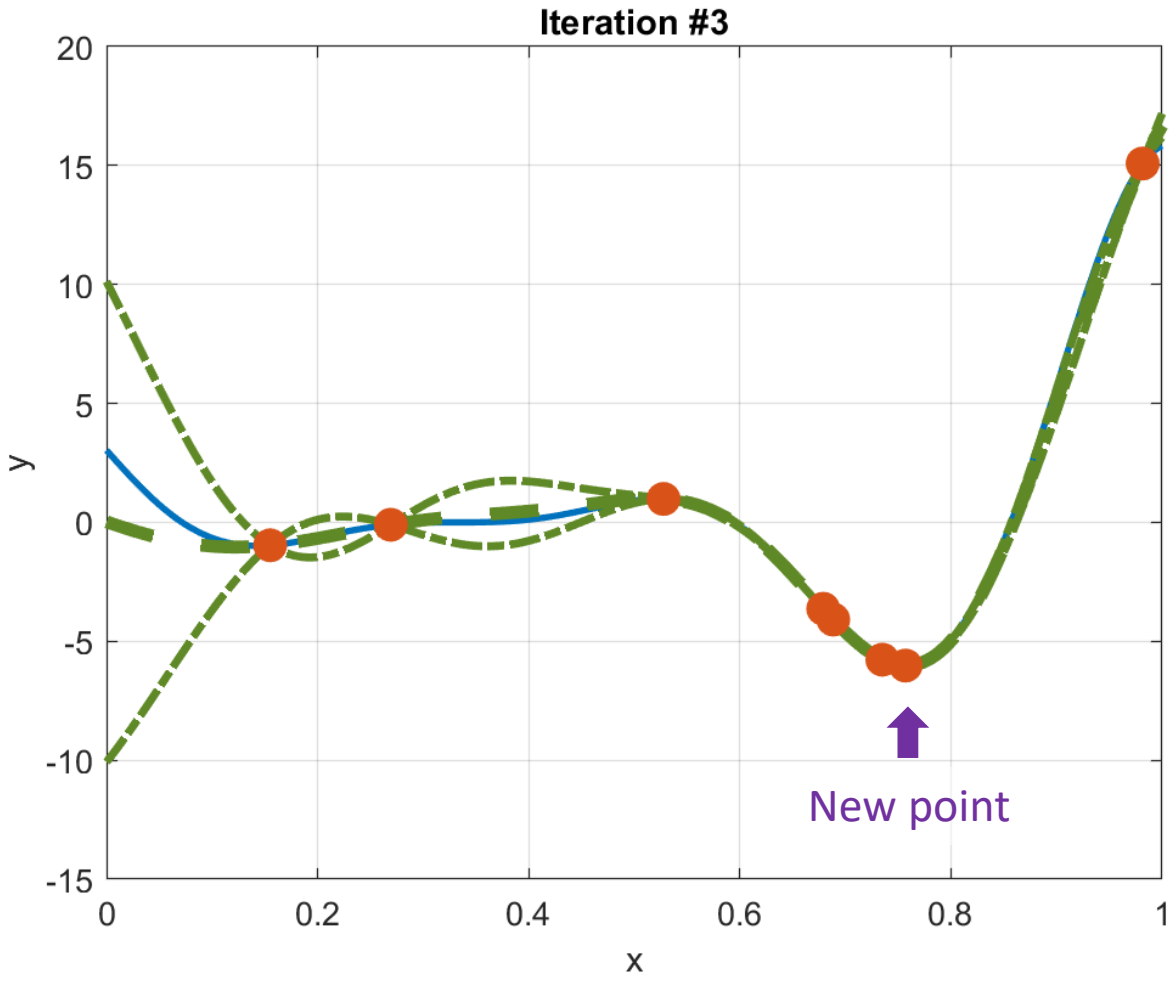
$$\mathrm{PoI}(\boldsymbol{x}) = \phi\left(\frac{y_{\min} - m(\boldsymbol{x})}{\sqrt{c(\boldsymbol{x},\boldsymbol{x})}}\right)$$

Cumulative distribution function (CDF) of standard normal distribution

Current minimum ($y_{\min}$)



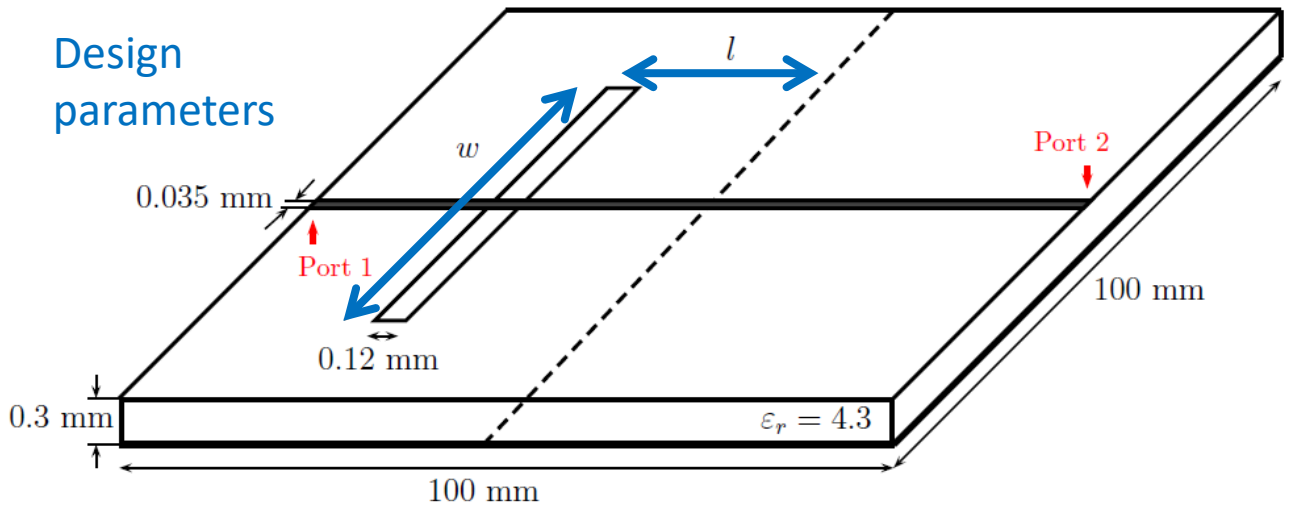
Iterative scheme

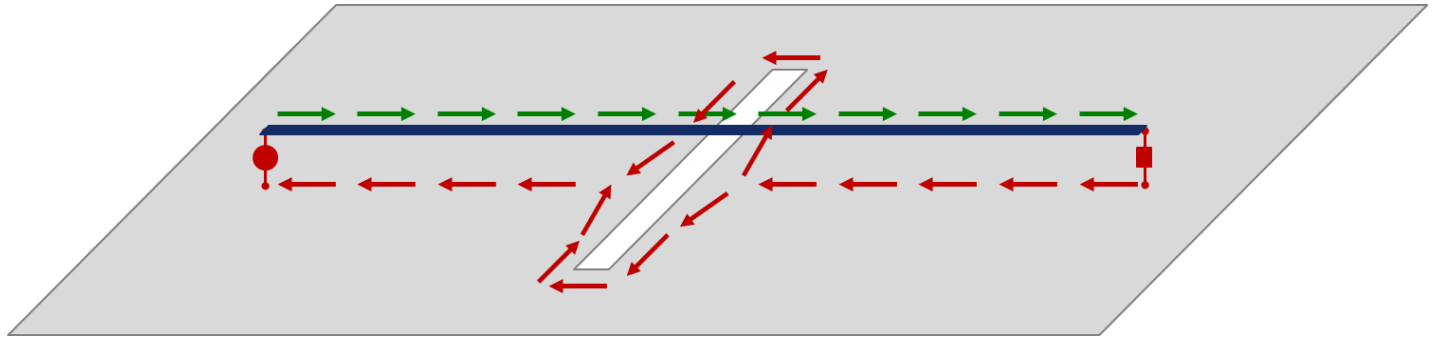


- Fast convergence to global minimum
- GPR model not required to be accurate everywhere

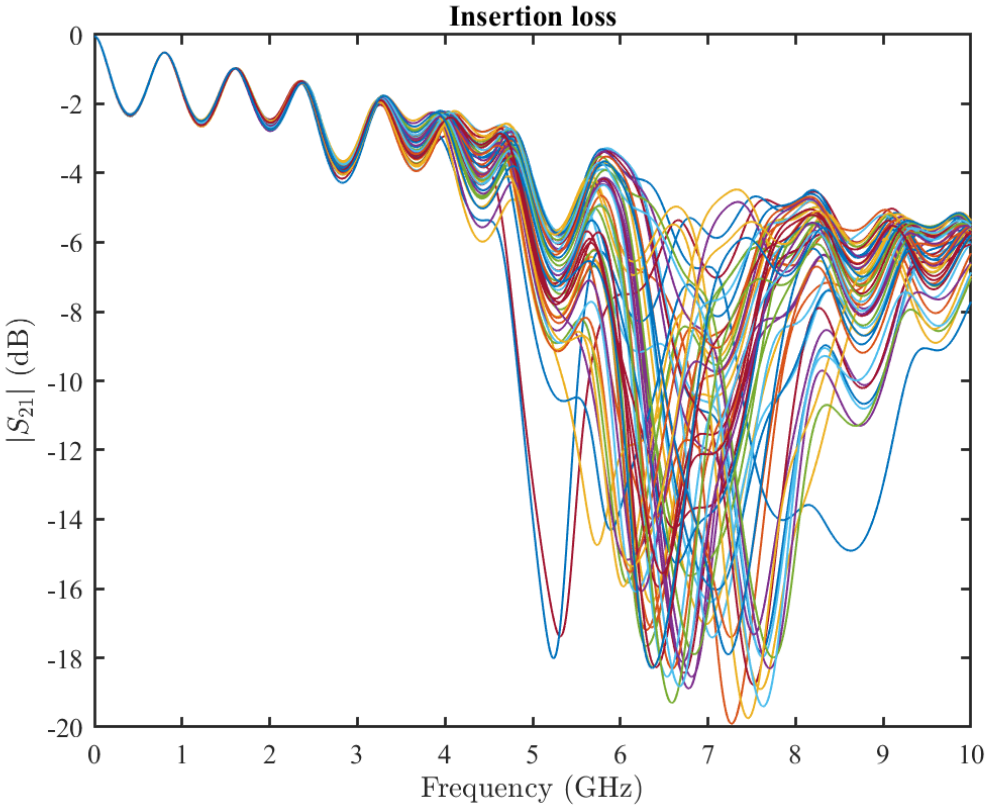
Application example: Microstrip line

Microstrip with ground plane discontinuity [3]



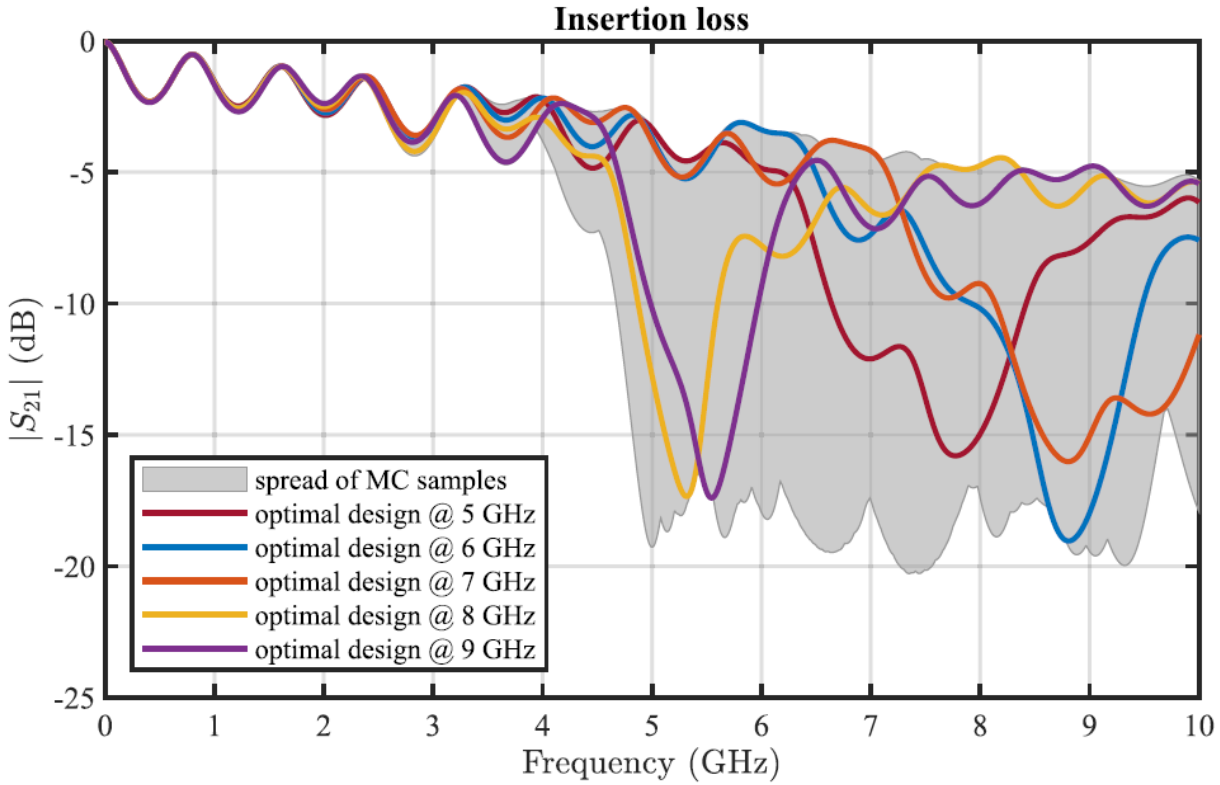


Goal 🎯 : find the values of **slot size** and **location** that minimize the **insertion loss** (maximize S21) at a given frequency



Optimization results

Optimal values at 5 GHz, 6 GHz, … 9 GHz



Optimization results for each frequency:

Frequency	f_0 (GHz)	5	6	7	8	9
Optimal value	l (mm)	11.5	13.7	17.6	15.5	19.0
	w (mm)	13.1	11.4	11.0	18.7	18.3

Only 30 simulations!

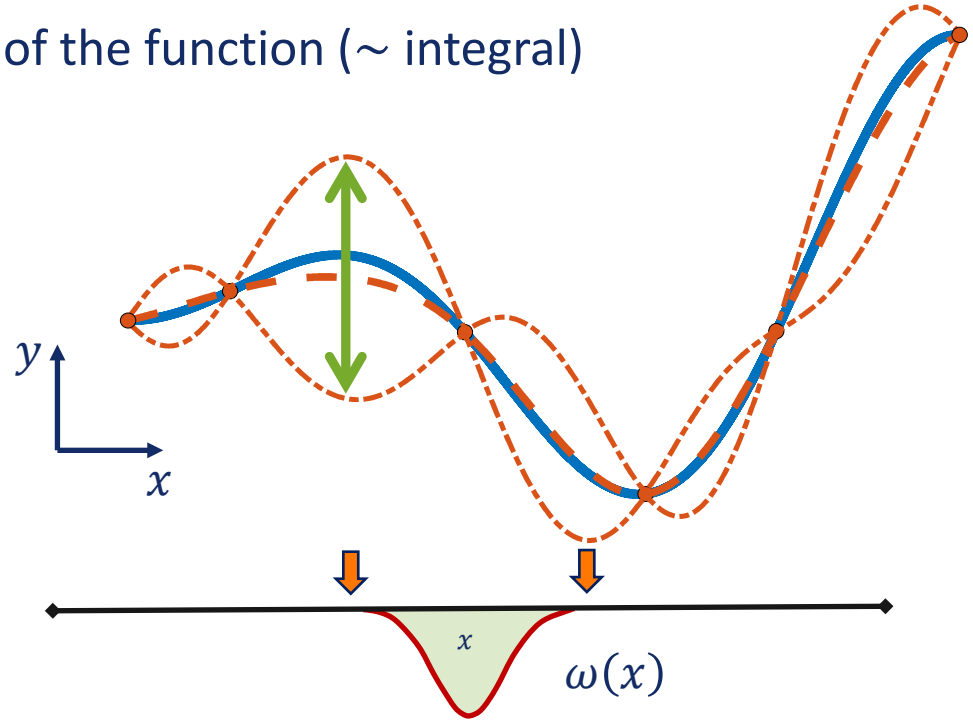
Active learning for UQ

AFs for BO may not be optimal in UQ settings

❌ They do not account for parameter distribution (which is assumed to be uniform within design range)

❌ They are optimized to reach the global minimum

Now the target 🎯 is a moment of the function (~ integral)

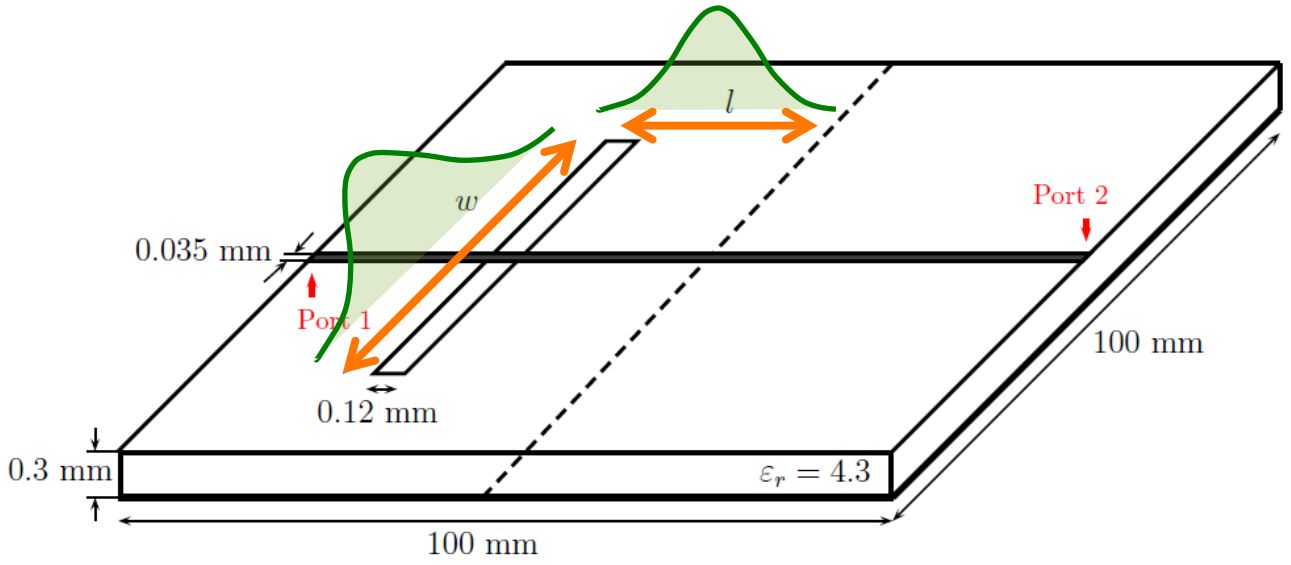


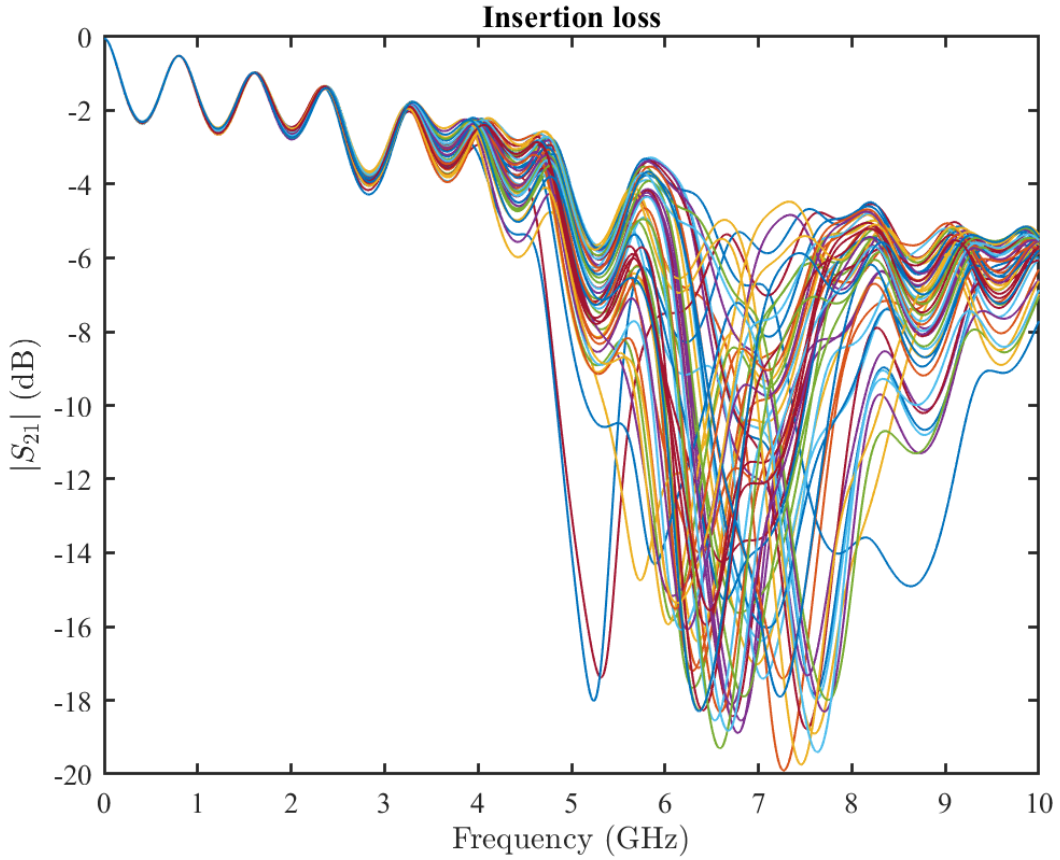
$$m_n = \int \left(\mathcal{M}(\boldsymbol{x})\right)^n \omega(\boldsymbol{x}) d\boldsymbol{x}$$

Application example

Let's consider again the microstrip line with ground plane discontinuity [4]

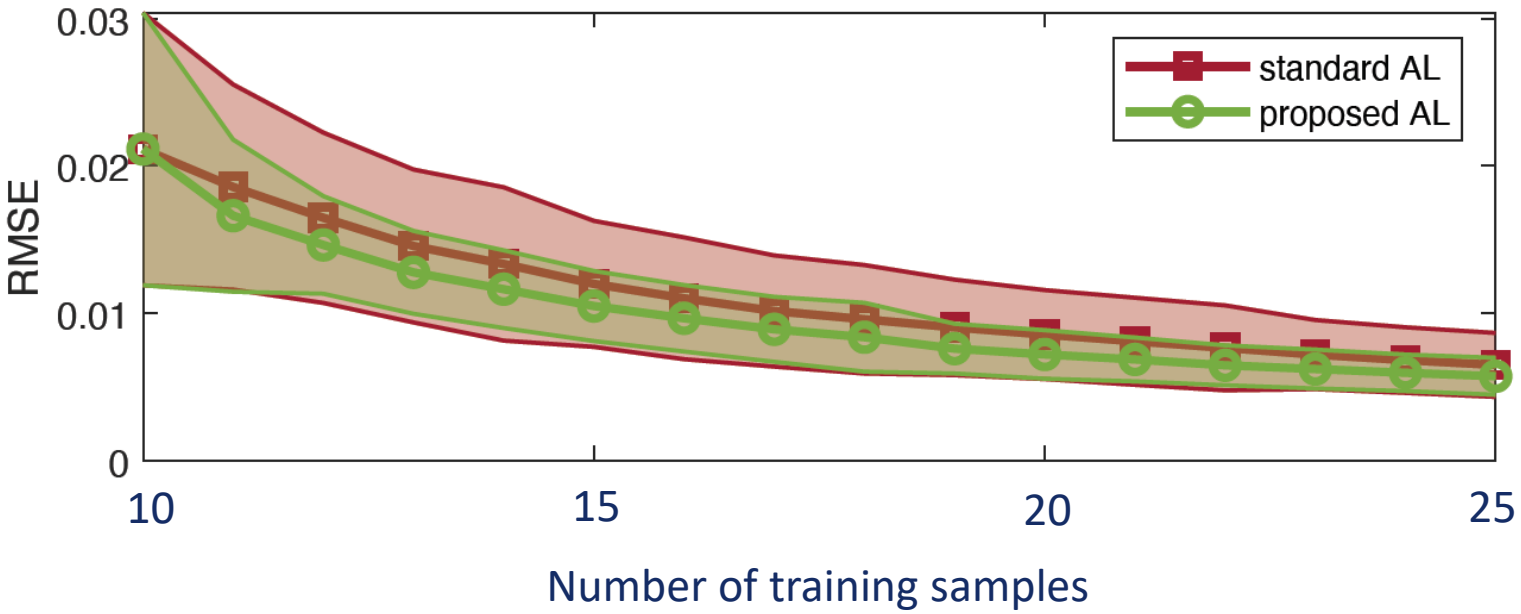
- We assign a probability distribution to w, l





Results

- We start from 10 training samples, and we query 15 additional samples
- We consider 50 independent runs
- We compare three different approaches:
  - Standard AL (deterministic): minimizes predictive variance
  - Stochastic AL: minimizes predictive variance of the mean



Take-home messages

- GPR allows a fast construction of surrogate models based on limited training data
- GPR additionally provides information on the prediction confidence
- Use of GPR surrogates is extended to UQ (with confidence information)
- (Corrections to the naïve GPR covariance may be required to improve confidence accuracy)
- Prediction confidence can be used to drive the iterative acquisition of training data in optimization and UQ

References

[1] P. Manfredi, “Probabilistic uncertainty quantification of microwave circuits using Gaussian processes,” IEEE Transactions on Microwave Theory and Techniques, vol. 71, no. 6, pp. 2360-2372, June 2023. DOI: 10.1109/TMTT.2022.3228953

[2] P. Manfredi, “Probabilistic uncertainty propagation using Gaussian process surrogates,” International Journal for Uncertainty Quantification, vol. 14, no. 6, pp. 71-104, 2024. DOI: 10.1615/Int.J.UncertaintyQuantification.2024052162

[3] P. Manfredi, “Conservative Gaussian process models for uncertainty quantification and Bayesian optimization in signal integrity applications,” IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 14, no. 7, pp. 1261-1272, July 2024. DOI: 10.1109/TCPMT.2024.3390402

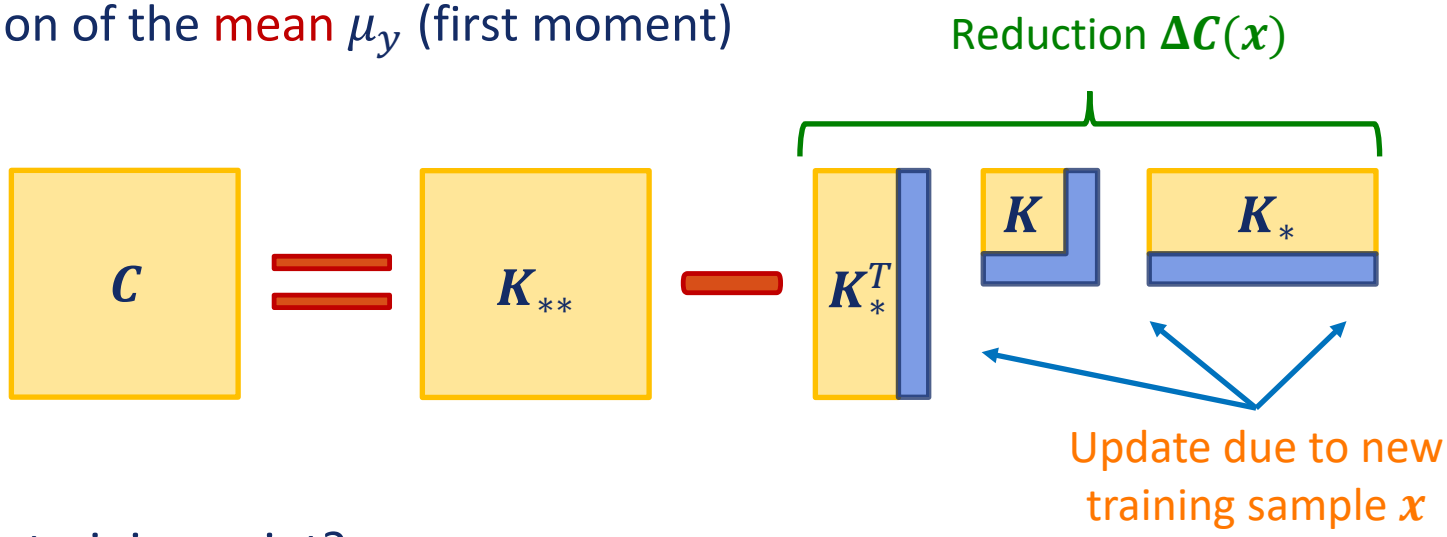
[4] M. Cusano, R. Trinchero, I.S. Stievano, S. Grivet-Talocia, P. Manfredi, and S. Schatt, “A multi-output active learning method for the uncertainty quantification of PCB lines,” 29th IEEE Workshop on Signal and Power Integrity (SPI 2025), Gaeta, Italy.

Stochastic active learning

Let us focus of the Monte Carlo prediction of the mean μ_y (first moment)

Its **predictive variance** reads:

$$\mathrm{Var}(\mu_y) = \frac{1}{N^2}\sum_{i,j=1}^{N} C_{ij}$$



How does it change when we add a new training point?

💡 Choose new training point such as the reduction $\boldsymbol{\Delta C}$ in the predictive variance of μ_y is maximized

✔ Inherently accounts for parameter distribution (μ_y is a statistical moment!)

✔ Does not depend on output values y ($\boldsymbol{C}$ only depends on input points x)

⇒ we can "test" multiple candidate training points without running actual simulations!!! 🎉

Stochastic active learning (cont'd)

Algorithm:

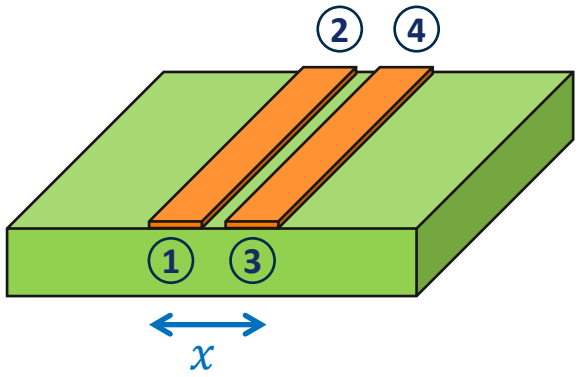
1. Explore candidate points and find the one that maximizes $\boldsymbol{\Delta C}$
2. "Label" (query) the new selected training point (= simulate the corresponding output y)
3. Re-train (update) the GPR model

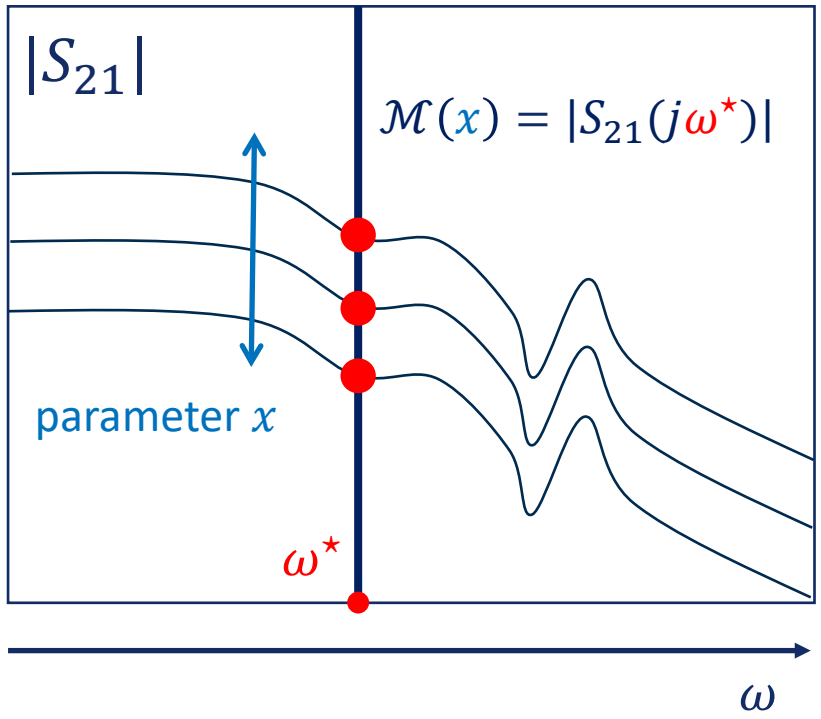
$$\boldsymbol{x}^{\star} = \underset{\boldsymbol{x}}{\operatorname{argmax}} \sum_{i,j=1}^{N} \Delta C_{ij}^{(\nu+1)}(\boldsymbol{x})$$

Iteration index

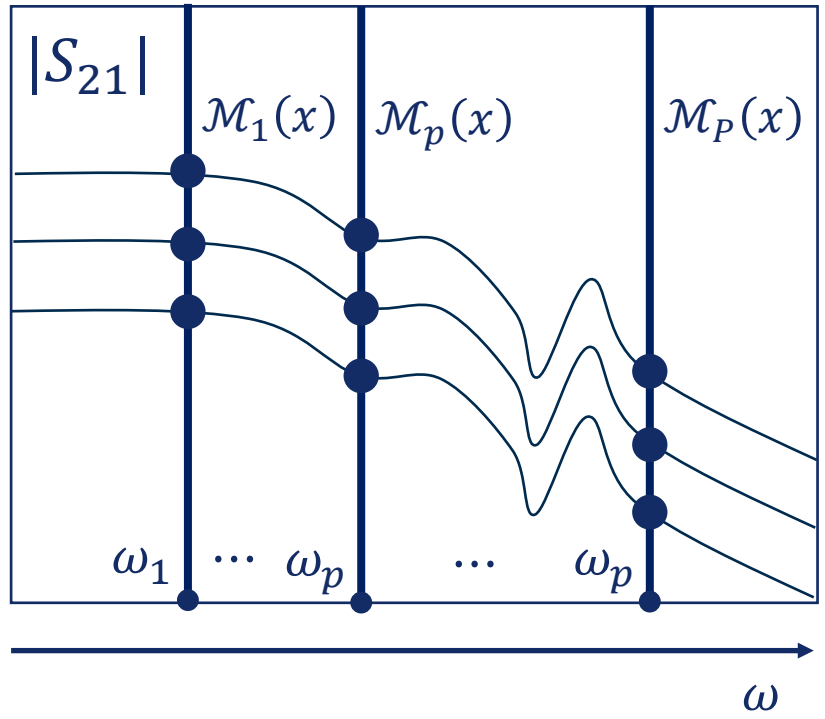
From single to multiple outputs

So far, we implicitly consider the output of interest y to be a scalar





E.g., magnitude of scattering parameter at a given frequency $\omega^\star$



How do we deal with multiple (say P) outputs?

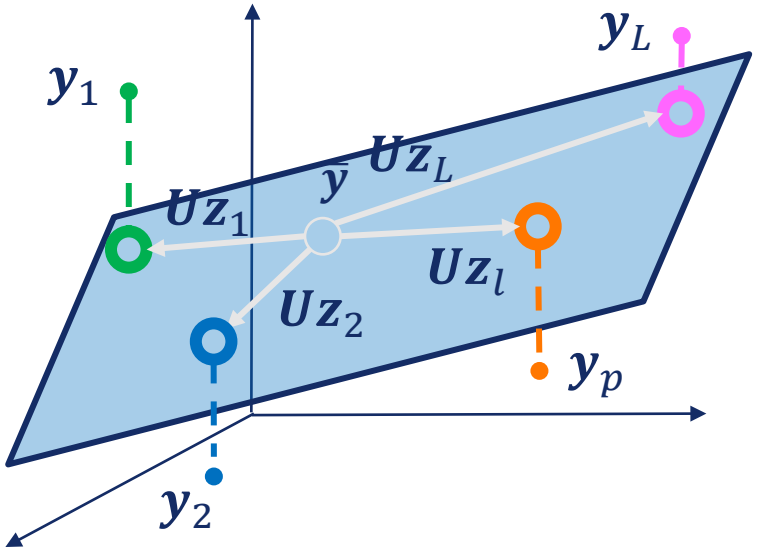
Principal component analysis (PCA)

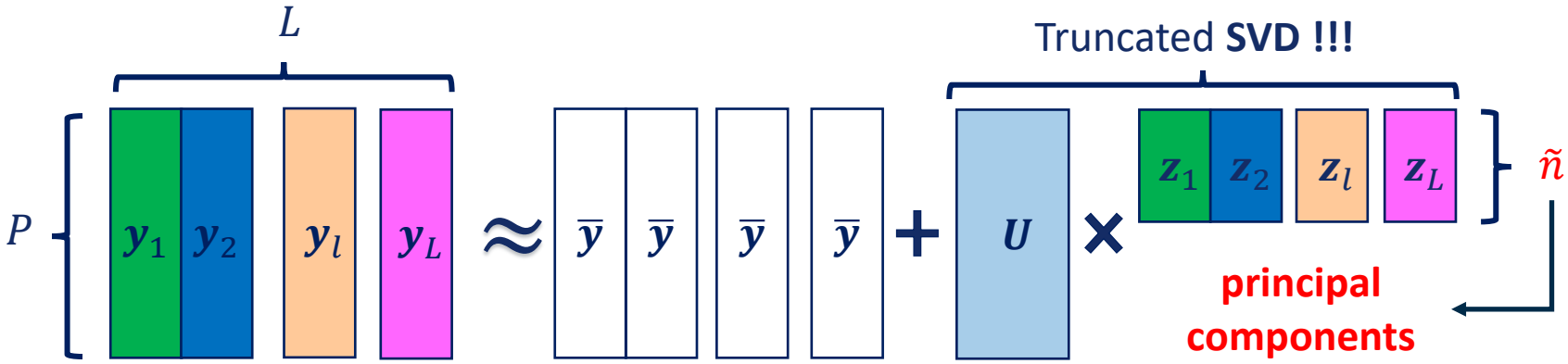
A naive approach would be to train P separate models, one for each output

❌ not very efficient if L and/or P is large!

In a problem with P outputs, the training dataset becomes a $P \times L$ matrix

Principal component analysis (PCA) is an effect tool to reduce output dimensionality





✔ Train a model for the $\tilde{n}$ "principal components" only

✔ Recover the original outputs thanks to PCA linearity

PCE-GPR Toolbox

🛠️ An open-source toolbox implementing the hybrid PCE-GPR method is available [6],[7]!

May 6, 2025 (v1.0.0) | Software | Open

View

PCE-GPR: A toolbox for high-dimensional uncertainty quantification

Manfredi, Paolo

This is the first official release of the PCE-GPR Toolbox, a MATLAB and UQLab-based toolbox for uncertainty quantification hybridizing Polynomial Chaos Expansion (PCE) and Gaussian Process Regression (GPR) methods. Features GPR-based surrogate modeling…

Uploaded on May 6, 2025

8 0

Software requirements

- MATLAB (2023b or newer)
- UQLab (version 2.0.0) – https://www.uqlab.com/
- Generalized chi-square distribution package (gx2) – https://github.com/abhranildas/gx2

[6] P. Manfredi, PCE-GPR: A toolbox for high-dimensional uncertainty quantification, May 2025, DOI: 10.5281/zenodo.15348860
[7] P. Manfredi, "Polynomial chaos vs kernel machine learning methods for uncertainty quantification: A comparative study and benchmarking with the hybrid PCE-GPR approach," Computer Methods in Applied Mechanics and Engineering, vol. 449, part A, article no. 118523, pp. 1-35, February 2026. DOI: 10.1016/j.cma.2025.118523